LA-UR 99-1233

QUARC: A Remarkably Effective Method for Increasing the OCR Accuracy of Degraded Typewritten Documents

Michael Cannon, Judith Hochberg, and Patrick Kelly
Los Alamos National Laboratory

1999 Symposium on Document Image Understanding Technology, Annapolis, Maryland

Mailing Address:
Michael Cannon
Mail Stop B265
Los Alamos National Laboratory
Los Alamos, NM 87544

QUARC: A Remarkably Effective Method for Increasing the OCR Accuracy of Degraded Typewritten Documents

Michael Cannon, Judith Hochberg, and Patrick Kelly
Los Alamos National Laboratory

Abstract. *We present a practical method for improving the OCR accuracy of degraded typewritten document images. Our method is based on a judicious selection of a restoration algorithm for each document that is to be processed. The selection is based on a comprehensive assessment of the quality of the document. The assessment quantifies the severity of a variety of document degradations, such as background speckle, touching characters, and broken characters. A statistical classifier then uses these measures to select an optimal restoration method for the document at hand. On a 41-document corpus, our methodology improved the corpus OCR character accuracy by 24% and the word accuracy by 30%.*

1. Introduction

Commercial OCR algorithms perform well on clean laser-written documents. However, many organizations have huge archives of typewritten material, much of it of marginal quality. For example, the U.S. Department of Energy has an archive of over 300 million classified documents consisting of typewritten documents, teletypewriter output, and carbon copies on aging fibrous paper. As part of the declassification review process, almost all of these documents have been photocopied and/or photoreduced. By today's OCR standards, this archive and others like it are of marginal quality. Even though many successful document enhancement methods are known [1 - 3], they must often be applied under human guidance to avoid further image degradation. Unsupervised use of enhancement software can lead to a marked degradation in corpus OCR accuracy[4].

In this paper we present an effective method for *automatically* selecting the optimal restoration method for each document in a corpus. The method consists of two parts. First, we use five measures to assess the quality of a document image. Second, we use this quality assessment to automatically select an optimal restoration algorithm for each document by means of a statistical classifier. After restoration, we show a marked improvement in corpus OCR accuracy. On a 41-member document corpus, our methodology resulted in a 24% improvement in OCR character accuracy and a 30% improvement in word accuracy. We call our procedure **QUARC**: **QU**ality **A**ssessment, **R**estoration, and o**C**r.

2. Data

The Department of Energy made a 41-member 300-dpi corpus of document images available to us for this work. Its quality is representative of the archive mentioned above. Ground-truth text files for the documents were also made available. We used Caere OmniPage Pro v8.0 to perform OCR and found the character accuracy of the corpus to be 65.72% and the word accuracy to be 49.01%.

3. Quality Measures

Our document image quality measures are designed to quantify the document degradations we observed in the DOE corpus. Many of these degradations are illustrated in Figure 1.

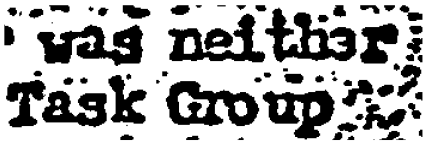

Figure 1. A portion of a page from the 41-member DOE corpus. This is a photocopy of a low-contrast carbon copy, which originally had neither background speckle nor broken characters. The photocopier automatically set a threshold to map subtle changes in gray tone to black or white.

We formulated five quality measures, each normalized to the range 0 to 1. The following is a preliminary description of the measures. A more technical definition will be given in the oral presentation of the paper and is also found in [5] and [6].

1. *Small Speckle Factor (SSF).* The small speckle factor measures the amount of black background speckle in the document image. The origin of the speckle varies. In our DOE corpus, much of it arises from photocopying low contrast documents (Figure 1). The background speckle can sometimes be so severe that it is interpreted as text by the OCR engine.

2. *White Speckle Factor (WSF).* Many degraded documents exhibit fattened character strokes. This problem can arise in carbon copies of documents, especially photocopies of carbon copies. The fattened stroke width can lead to OCR difficulties by creating unexpected small white connected components or by reducing or eliminating expected white components.
3. *Touching Character Factor (TCF).* The touching character factor measures the degree to which neighboring characters touch. Like white speckle, touching characters are caused by fattened strokes, as seen in the word "was" in Figure 1. Touching characters cause problems for OCR by making it difficult to differentiate between certain letters such as "ni" and "m", and by creating completely novel and uninterpretable text.
4. *Broken Character Factor (BCF).* The broken character factor measures the degree to which individual characters are broken. In our 41-document corpus, broken characters are the largest single cause of OCR errors. Broken characters often arise from photocopying low contrast documents, as seen in both occurrences of the letter "e" in Figure 1.
5. *Font Size Factor (FSF).* We find a correlation in our corpus between OCR accuracy and the size of the font. This correlation might not stem from the font size *per se*, but rather from degradations that accompany an increase or decrease in the size of the font.

As we developed the five quality measures, some of the parameters within each measure were tweaked in order to make the correlation with the OCR error as high as possible[5]. The correlation between quality measures and the OCR error was sufficiently high to motivate us to attempt to predict the OCR error rate based on the quality measures themselves. The prediction was based on a linear combination of the quality measures and was computed using a least square method[6]. We obtained the weights for the linear combination by training on half the data; we then used the weights to predict the error rates for the other half. The correlation between the actual OCR error rates and the predicted ones was .89, an indication that the quality measures are indeed meaningful.

4. Restoration Methods

Our document image restoration methods are designed to repair the degradations reflected in the quality measures. We implemented fourteen restoration algorithms, but determined that only four were effective[5]. We applied each restoration method to the documents in our corpus, OCR'd all the resulting document images, and then computed the corresponding OCR accuracies. The restored version of a document with the highest OCR accuracy indicated the restoration algorithm that was best suited for that particular document.

- *Do Nothing*. It may be that the best enhancement for a document image is to leave it alone. Doing nothing is therefore included in our suite of restoration algorithms.
- *Cut on Typewriter Grid*. The documents in our corpus lie on a fixed-width typewriter (or teletypewriter) grid. If a document is plagued with touching characters, we should in principle be able to separate them if the typewriter grid is known. Our method for determining the typewriter grid is an extension of a method put forth by Lu [7]. We find the typewriter grid by first computing the Fourier transform of the vertical projection of lines of text. The average of the magnitude-squared of the transforms is computed. A typical average is shown in Figure 2. The prominent peak indicates the period of the typewriter grid.

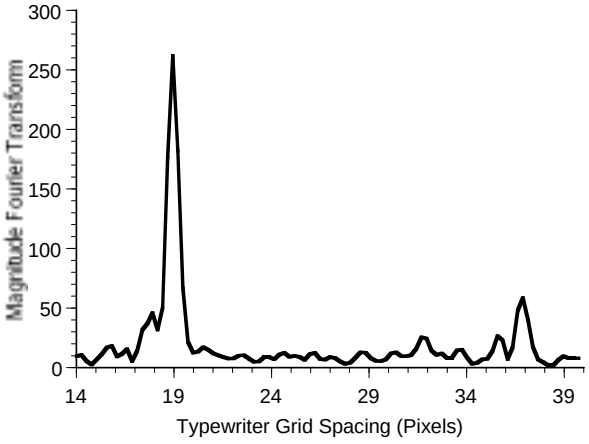

Figure 2. An average of the Fourier transform magnitudes of several lines of text. The prominent peak at 19 pixels indicates the width of the typewriter grid.

Our cutting algorithm moves along each line of text extracting two neighboring characters at a time. The location of the characters is known from the typewriter grid. We check to see if the two characters constitute the same black connected component - if they do, a white vertical line is drawn between them. If the characters do not touch, the line is not drawn, as it may destroy character detail such as serifs. In order to stay synchronized with the true character positions, the algorithm frequently computes the cross correlation between the typewriter grid and the vertical projection of the line of text and adjusts its position to the point of maximum correlation. In the Appendix we describe the special case of a *variable-width* typewritten font.

- *Global Fill Holes and Breaks*. In order to fill in breaks and fractures in characters, we employ a method described by Loce and Dougherty [8]. The filling operation consists of operating on the document image with 8 simple morphological kernels and ORing the results together.
- *Global Despeckle*. In order to suppress black background speckle while preserving character shape, we rely on another method described by Loce and Dougherty [9]. They prescribe a union of a 2-erosion basis set. Each kernel is 3x3 with two nubbins on it, which we apply globally to the document image.

5. Automatic Restoration Method Selection

We are now in a position to train the statistical classifier that will predict the best restoration method for new documents. We know each document's five quality measures that will be input to the classifier. We also know the best restoration method (out of the four-method set) that will optimally improve it. More generically, we have 41 objects, each described by five features and belonging to one of four classes, a classic pattern classification problem. We therefore trained a statistical classifier, using the Pocket algorithm [10], to assign each document, based on its five quality measures, to one of the four restoration methods. We did this two times, training first to the best category for improving OCR *character* accuracy, then to the best category for improving *word* accuracy.

We tested the statistical classifier using cross-validation. That is, we cycled through the entire corpus, on each iteration training on 40 documents and testing on the 41st. The OCR improvement resulting from the best possible restoration method for each document gave an upper bound for these results. For a lower bound, we found the outcome of choosing a restoration method randomly.

The following subsection presents the results of the cross-validation test in three ways: according to improvement in OCR character accuracy, improvement in OCR word accuracy, and selection of the optimal OCR algorithm. By all measures, the method was a success.

5.1 OCR Character Accuracy Results

As shown in Table 1, automatic selection of a restoration method substantially improved the OCR character accuracy of the corpus. The character accuracy in the 41-document subcorpus increased from 65.72% to 81.18%, a hefty 24% improvement. The improvement was not quite as good as our established upper bound (the outcome using the best restoration method for each document), but certainly better than our lower bound (from random selection of a restoration method).

Restoration Selection Procedure	Character Accuracy	Word Accuracy
No restoration	**65.72%**	**49.01%**
Random selection of four restoration methods	**72.52%**	**53.76%**
Statistical classifier selection of four methods	**81.18%**	**63.80%**
Best of four restoration methods	**82.51%**	**65.62%**

Table 1. A compilation of OCR character accuracies resulting from a variety of restoration method selection criteria.

5.2 Restoration Cascade

It is tempting to couple our restoration methods in pairs. Perhaps a best restoration method would consist of background despeckle followed by a cut on the typewriter grid. We have experimented with some of these combinations and obtained spotty results. Some restoration cascades improved four or five of our documents by an additional 10% or so character accuracy. But in general, we saw little improvement in the corpus OCR accuracy as a whole.

One reason for this lackluster result may be that some of our restoration methods tend to be dual purpose already. For example, the Loce/Dougherty despeckle algorithm also tends to thin fattened strokes, as shown in Figure 3. The algorithm by itself has the effect of a cascade. Another reason the cascade

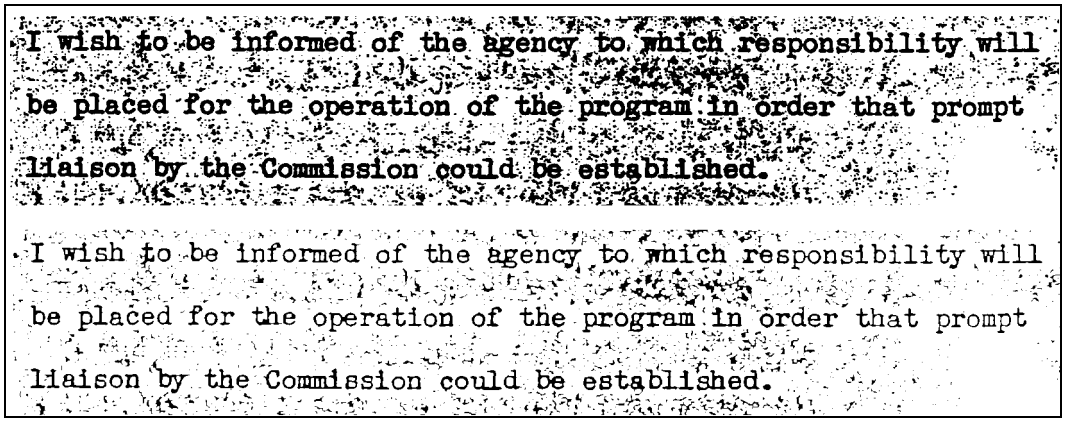

Figure 3. Top: a portion of an original document plagued by background speckle and fattened stroke widths. Bottom: the same portion of the document after enhancement by the Loce/Dougherty 2-erosion basis set. Note that both degradations have been addressed by the one enhancement method.

does not work is that the restoration methods also introduce artifacts into the document image. Perhaps the application of two methods in cascade introduces too many artifacts for the OCR engine to handle, and the benefit of the restoration methods is lost.

We believe that a cascade of restoration methods may still have merit; it is just too difficult to show it and train a classifier accordingly on our 139-member corpus. We will investigate the approach further when a larger 1000-member document corpus becomes available to us.

6. Software Implementation

Our entire approach to document image restoration and OCR has been implemented in three C++ software modules.

1. **TRAIN**: The user runs **TRAIN** in a directory containing many document images and their ground truth text files. **TRAIN** restores each image using four different methods and then computes the OCR accuracy resulting from each one. The quality measures from each document image as well as its best restoration method are written to a disk file. It takes several hours for **TRAIN** to run.

2. **CLASSIFIER**: The user next runs a program called **CLASSIFIER**, which reads in the disk file created by **TRAIN**. **CLASSIFIER** uses this information to create the classifier that is used by **QUARC** to automatically select an optimal restoration method based on a document's quality measures. It takes just a few seconds for **CLASSIFIER** to run; information defining the classifier is written to a disk file.

3. **QUARC**: The main production program is **QUARC**, which reads in the disk file produced by **CLASSIFIER** and then proceeds to optimally restore and **OCR** the document images that are passed to it.

7. Conclusions

We have presented a successful method for automatically improving the quality of document images in a typewritten archive, and we demonstrated a marked increase in OCR accuracy. The 24% improvement in OCR character accuracy and the 30% improvement in word accuracy on our 41-member corpus are significant. The method is easy to use - we view it as a pre-OCR cleanup operation, and it takes about one-tenth the computational effort of the OCR process itself. We like the automatic classifier because it takes into account all five quality measures when selecting an appropriate restoration method, rather than using one or two thresholds set by trial and error on a subset of the measures. Our methodology is not limited to our suite of four restoration methods. Any other restoration methods can be included, even if they are folded into the OCR process itself, as long as they are "best" for a meaningful number of documents in a training corpus.

On the other hand, the need to train a classifier for best performance on a particular corpus is a real effort, because it requires textual ground truth. Perhaps one-time training on a very large corpus would obviate the need for repeated training on smaller specialized corpora.

Acknowledgments

Professor Nathan Brener and his staff at Louisiana State University scanned the 41 documents used in this study and provided the keyed-text ground truth. Don Hush at Los Alamos provided assistance with the statistical classifier and the Pocket algorithm. Tom Curtis, Department of Energy, and Jim Campbell and Gary Craig, Federal Intelligent Document Understanding Laboratory, provided important administrative and technical support. Technical conversations with Becker Drane and Steve Dennis,

Department of Defense, were particularly valuable to us.

References

1. Victor T. Tom and Paul W. Baim, *Enhancement for Imaged Document Processing*, Proceedings 1995 Symposium on Document Image Understanding Technology, Annapolis, MD, p154.
2. P. Stubberud, et. al, *Adaptive Image Restoration of Text Images that Contain Touching or Broken Characters*, Proceedings ICDAR'95 Third International Conference on Document Analysis and Recognition, Montreal, 1995, p778.
3. TMSSequoia, "ScanFix Software," 206 West 6th Avenue, Stillwater, OK, 74074 ©1997.
4. In a joint experiment with Highland Technologies on a 140-document corpus, we found the OCR character accuracy dropped from 80% to 66% after applying ScanFix corrective techniques in an unsupervised manner.
5. T. M. Cannon, J. G. Hochberg, P. M. Kelly, *Quality Assessment and Restoration of Typewritten Document Images*, submitted to International Journal on Document Analysis and Recognition, expected publication date: Spring, 1999.
6. Michael Cannon et. al, *An Automated System for Numerically Rating Document Image Quality*, Proceedings 1997 Symposium on Document Image Understanding Technology, Annapolis, MD, p162.
7. Yi Lu, *On the Segmentation of Touching Characters*, Proceedings, ICDAR'93 Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, 1993, p440.
8. Robert P. Loce and Edward R. Dougherty, *Enhancement and Restoration of Digital Documents*, SPIE Optical Engineering Press, 1997, p192.
9. ibid., p198.
10. S. I. Gallant, *Preceptron-based Learning Algorithms*, IEEE Trans. Neural Networks, Vol. 1, No. 2, 1990, p179.

Appendix

In Section 4, we describe a method for cutting touching characters on a typewritten grid. Since not all typewritten fonts are fixed-width, it is important to determine if we are dealing with a variable-width font before cutting on a non-existent fixed-width grid. We can determine if a document has a fixed-width font by measuring the height of the peak shown in Figure 2. If the height is more than ten standard deviations above the mean of the transform, the document has a fixed-width font, otherwise the font is variable-width and no attempt is made to separate touching characters.